

손 제스처를 이용한 조이스틱 방식의 마우스제어 방법

정진영[†], 김정인^{††}

A Joystick-driven Mouse Controlling Method using Hand Gestures

Jin-Young Jung[†], Jung-In Kim^{††}

ABSTRACT

PC users have long been controlling their computers using input devices such as mouse and keyboard. To improve inconveniences of these devices, the method of screen-touching has widely been used these days, and devices recognizing human gestures are being developed one after another. For example, Kinect, developed and distributed by Microsoft, is a non-contact input device that recognizes human gestures through motion-recognizing sensors, thus replacing the mouse as an input device. However, when controlling the mouse on a large screen, it suffers from the problem of requiring large motions in order to move the mouse pointer to the edges of the screen. In this paper, we propose a joystick-driven mouse-controlling method which enables the user to move the mouse pointer to the corners of the screen with small motions. The experimental results show that movements of the user's palm within the range of 30 cm ensure movements of the mouse pointer to the edges of the screen.

Key words: Mouse Controlling, Gestures, Kinect, Joy-Stick, Recognition

1. 서 론

컴퓨터 사용자들은 입력장치로서 마우스와 키보드를 주로 사용해 왔다. 사람들은 오랜 시간 마우스와 키보드를 사용하면서 클릭은 누르는 것, 드래그는 누른 뒤 옮기는 것과 같은 방식에 적응하여 컴퓨터를 제어하는 것을 당연시하게 되었다. 이러한 사용자 인터페이스는 스마트폰으로 이어져서 클릭, 드래그, 스와이프, 멀티터치 등에서 손가락을 이용한 터치 인터페이스를 갖추게 되었고, 마찬가지로 사람들은 그러한 인터페이스에 적응되어 편하게 사용하고 있다.

키보드와 마우스에 이어서 터치가 인터페이스의 대세로 자리매김하였으나, 최근 비접촉식 입력장치

도 많은 사람들의 관심을 받고 있다. 마이크로소프트 사는 2010년 11월에 사용자의 모션을 인식하는 키넥트(Kinect)를 출시하였다. 키넥트는 키네틱(Kinetic)과 컨넥트(Connect)의 합성어로 사람의 동작을 인지하여 컴퓨터 시스템에 연결하는 동작인식센서 장치이다. 처음 출시된 XBOX-360의 디바이스로 출시되어 주로 게임 콘텐츠에 많이 이용되었지만, 2012년 2월에 게임용도가 아닌 임베디드 기기에서 사용할 수 있는 Kinect for Windows를 출시하여 Windows 운영체제 기반의 기기에서도 쉽게 이용할 수 있게 되었다[1]. 그 이후로 여러 가지 모션인식 게임과 응용소프트웨어가 함께 개발되어졌고, 현재까지도 키넥트의 다양한 활용 방안에 대한 연구가 꾸준히

* Corresponding Author : Jung In Kim, Address: (608-711) Dept. of Computer Engineering, Tongmyong University, Sinsun-ro 428, Nam-gu, Busan, Korea, TEL : +82-51-629-1174, FAX : +82-51-629-1169, E-mail : jikim@tu.ac.kr

Receipt date : Dec. 29, 2014, Revision date : Nov. 30, 2015
Approval date : Dec. 1, 2015

[†] Department of Computer Engineering, Tongmyong University (E-mail : jeous88@naver.com)

^{††} Department of Computer Engineering, Tongmyong University

* This work (Grants No. C0135691) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2013

진행되고 있다.

본 연구에서는 키넥트 SDK에서 제공되는 스켈레톤 트래킹을 이용하여 손동작을 인식하는 것으로 컴퓨터의 마우스를 제어한다. 특히, 동작인식은 팔의 제스처를 이용하기 때문에 사용자들은 쉽게 피로해 지거나 어색할 수 있는데 그러한 문제점을 줄이기 위해 작은 움직임에도 마우스 포인터가 원하는 곳까지 쉽게 이동 할 수 있는 방법에 대해서 연구하였다.

2. 관련연구

키넥트를 활용하는 비접촉식 인터페이스에 관한 연구는 국내외에서 많이 보고되고 있다. 국내의 키넥트 관련연구로는 윤영준 등은 키넥트의 RGB영상, 깊이영상, 인체 관절의 3차원 위치 정보를 이용하여 키보드, 마우스, 레이저, 폰을 대신하여 프레젠테이션 화면을 제어하는 연구를 하였다. 키넥트로 관절 정보를 활용하여 동작의 정확도를 높였고, 깊이 이미지로부터 입력된 영상에서 일정거리에 있는 영상만 출력하도록 하여 손을 인식시켰다[2].

김성민 등은 키넥트의 RGB-Depth 카메라를 활용하여 사람의 키를 추정하는 방법을 제안하였다. 키넥트 센서로부터 사람의 머리와 발까지의 거리를 각각 측정하고 키넥트와 사람 간의 수평거리, 키넥트의 지면에서부터의 높이를 측정하여 키를 추정하였다[3].

이민규 등은 키넥트에서 인식하는 손의 움직임 정보를 인식하여 키보드, 마우스와 같은 기존의 입력장치를 대신하기 위한 연구를 진행하였다. 키보드 모드는 가상의 버튼들을 배치하고 버튼에 일정시간동안 손의 위치 정보가 있을 때 이벤트를 발생시켰고, 마우스 모드에서는 포인터 조작과 클릭을 하였다[4].

윤신요 등은 키넥트를 이용하여 양손의 좌표를 추적하여, 그 두 좌표를 사용한 양손 마우스 인터페이스를 구현하였다. 커서의 제어는 오른손을 추적하는 함수를 이용하였고, 왼손으로는 우클릭, 좌클릭 이벤트를 발생시켰다[5].

전인배 등은 키넥트의 깊이센서를 이용하여 손 영역을 검출하고, 마우스를 구현하였다. 깊이센서로 얻은 데이터와 레이블링 알고리즘을 사용하였고 빛의 영향을 받지 않으며 처리하는 데이터 량이 많지 않아 처리속도가 빠르고 높은 인식률을 얻었다고 보고하였다[6].

김정인은 웹 카메라를 이용하여 사람의 손동작을

마우스 입력으로 대체하였다. 단순히 손의 위치를 인식하여 마우스 포인터를 인식시키는 것이 아니라, 마우스의 위치를 가장자리까지 원하는 곳으로 편리하게 위치시키는 효과적인 방법을 제시하였다[7].

해외의 관련연구로 Rajaganapathy는 언어 장애인을 위해 키넥트를 이용하여 수화를 번역 할 수 있는 방안에 대해 연구 하였다. 사람의 제스처에 대한 골격 위치를 습득하여 미리 정의된 제스처 입력값과 일치하면 해당되는 단어가 출력된다[8].

Andrea Sanna는 키넥트 센서를 이용하여 쿼드콥터를 제어하는 방법에 대해 연구하였다. 사용자의 제스처와 손을 이용하여 쿼드콥터를 더 직관적으로 운용 할 수 있다[9].

Yusman Azimi Yusoff는 시각화 의료기기에서 사용되는 2D 사진과 3D 그래픽을 키넥트를 이용하여 회전, 크기확대, 축소, 밝기조절, 슬라이드 제스처를 구현하였다[10].

이와 같이 손동작을 인식하는 연구는 꾸준히 진행되고 있으며, 특히 키넥트는 비접촉식 인식이라는 장점이 있어서 차세대 입력장치로서 크게 주목받고 있다는 것을 알 수 있다.

3. 키넥트를 활용한 마우스제어 방법

3.1 키넥트 센서를 이용한 일반적인 마우스 제어

키넥트(Kinect)는 2012년에 마이크로소프트사에서 출시한 센서장치이다. 이 센서는 사용자의 신체를 인식하는 기술과 깊이 영상 추출 기술, 음성인식 기술을 가지고 있어 여러 기술을 하나의 기기로 구현할 수 있다는 장점이 있다.

키넥트의 내부 구조는 Fig. 1과 같다. 사진의 가장 왼쪽엔 적외선 입자를 전방으로 쏘아주는 적외선 광

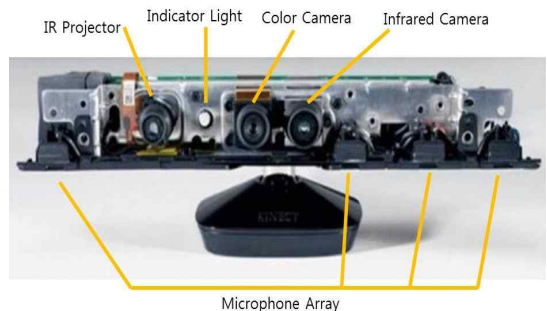


Fig. 1. Inner Structure of Kinect Sensor.

원/적외선 프로젝터가 있고, 그 오른쪽에 키넥트의 동작 상태와 이상여부를 알려주는 LED 인디케이터 (Indicator), 중앙에는 RGB 데이터를 얻기 위한 카메라가 있고, 맨 오른쪽에는 깊이 데이터를 얻기 위한 적외선 카메라가 있다. 센서의 시야 범위는 57도, 상하 시야각은 43도, 동작의 깊이 인식 범위는 1.2m~3.5m 이고 RGB 카메라는 최대 1280×960의 해상도를 제공하며 깊이 카메라는 최대 640×480의 해상도를 제공한다. 그리고 아래쪽에 있는 4개의 마이크 배열이 음성 인식을 담당한다. 키넥트의 머리 부분과 받침대 사이에는 모터가 탑재되어 있어 머리 부분의 상하 움직임을 제어 할 수 있어, 필요에 따라 위치를 보정 할 수 있다[11]. 이 센서를 소프트웨어적으로 컨트롤하기 위해 Microsoft에서는 개발도구인 SDK를 제공하고 있다.

사물의 깊이를 인식하는 기술은 키넥트에 내장된 적외선 프로젝터와 적외선 카메라로 인해 구현된다. 적외선 프로젝터에서는 픽셀 단위의 적외선 입자들을 전방으로 쏘아주고 전방의 물체에 반사된 적외선 데이터들을 적외선 카메라 센서에서 받아들여 깊이를 감지한다. 받아들인 데이터는 소프트웨어가 자동으로 주변의 명도조건과 비교하여 전방의 장애물에 대한 인식을 하고, 적외선 입자들에 대한 패턴 왜곡을 계산하여 포착된 사람과 주변 사물, 장애물에 대한 구별을 한다.

우리는 손동작으로 마우스를 제어하는 방식을 구현하기 위하여 스킴레톤 트래킹을 이용하였다. 스킴레톤 트래킹은 키넥트에서 사람의 관절 20개의 위치를 인식하는 기술이다. 편의상 본 연구에서는 필요한 오른손만 인식시켰다. 아래 Fig. 3과 같이 키넥트가

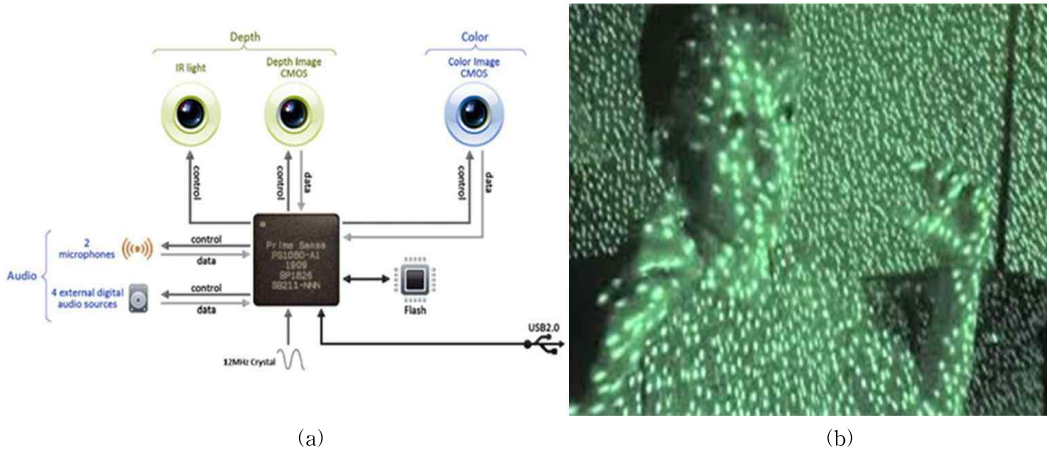


Fig. 2. (a) Structure About Kinect' s Color, Depth and Audio Sensor (b) Infrared Light of Kinect[12].

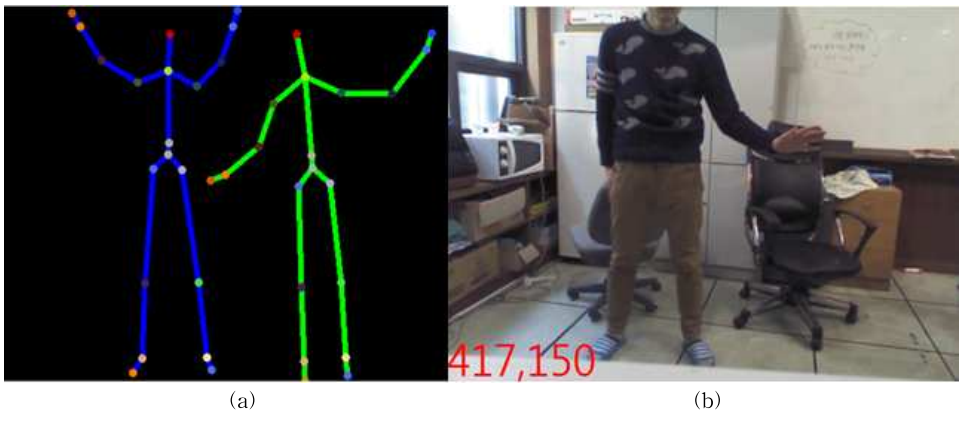


Fig. 3. (a) Kinect skeleton (b) Right hand recognition test.

오른손만 인식하여 화면의 어느 좌표에 손이 위치하여 있는지 나타내준다[13].

화면의 손 위치를 마우스의 위치로 대응시키는 방법은 어렵지 않다. 마우스 포인터가 키넥트 카메라의 손 움직임과 동기화되어 움직이게 하기 위해 C#의 User32.dll의 mouse_event 함수를 이용한다. User32.dll은 주로 사용자의 입력을 받아 WinAPI를 사용할 수 있게 해주는 파일로 마우스의 제어 기능을 포함하고 있다. 카메라의 해상도에서의 손 좌표값을 mouse_event 함수에 넣어주는 것으로 카메라 해상도의 손의 좌표가 화면 전체 해상도에서의 마우스의 좌표값이 된다. 이러한 방식으로 마우스를 제어하게 되면 화면에서 손이 움직이는 만큼 마우스 포인터가 움직이는 일반적인 마우스 위치제어가 가능하다.

다만 미세한 움직임에도 마우스 포인터의 떨림 현상이 발생한다든지, 주먹을 쥐는 행위의 클릭 이벤트 발생 시에 좌표 값의 오류가 나타난다든지, 대화면의 경우 가장자리까지 마우스포인터를 이동시키기 어려운 점 등은 개선되어야할 사항들이다.

하지만 단순한 손-마우스 동기화 방식을 사용하여 마우스를 제어할 때는 마우스 포인터를 스크린의 양쪽 끝이나 구석자리로 위치시키기 위해 사용자의 손동작이 상당히 커져야 한다. 이러한 점은 많은 사람들이 모여 있는 장소에서 사용하게 될 경우 상당히 어색하게 비춰질 수 있고 오랜 시간 사용할 경우 팔이 아프거나 힘들어질 수 있다. 또한, 마우스 이동이 필요 없는 상태인데도 돌아오는 손의 움직임을 인식하여 오동작하는 경우가 자주 발생한다. 이러한 문제점들을 해결하기 위해 비교적 작은 동작으로 화면의 어느 위치든지 빠르게 마우스 포인터를 보낼 수 있는 조이스틱형의 마우스 제어방식을 제안한다.

3.2 조이스틱형 마우스 제어 방법

조이스틱형 마우스 제어는 특정 중심점을 미리 결정해 두고, 또 하나의 변화점 위치 X를 짧은 시간 단위로 인식하면서 중심점과 변화점의 위치를 이용하여 실제 마우스의 위치를 계산해내는 방식으로 이루어진다. 두 점의 좌표 값을 이용하는 간단한 수식으로 방향과 속도 계산이 가능하며 계산되어진 새로운 좌표 값을 event_mouse 함수에 넣어주면 마우스의 현재 위치가 되는 것이다.

조이스틱형 마우스 제어방식 실험을 위하여 키넥

트가 인식하고 있는 손의 위치를 X 모양으로 화면에 표시하여 카메라 센서를 통해 출력되어지는 영상 화면에 나오도록 하였다. 그리고 영상 화면의 정중앙에 중심점을 찍고 그 점을 중심으로 반지름 50픽셀의 작은 원을 그려 넣어 그 원을 경계선으로 안과 밖을 구분 짓도록 하였으며, 실행 결과는 Fig. 4와 같다. 작은 원의 중심점과 변화점 X로 표시된 손의 위치 좌표 간의 거리를 구하여 50픽셀 보다 작으면 원의 안쪽, 크면 원의 바깥쪽을 나타낸다.

변화점 X의 위치가 중심원의 안쪽에 위치 할 때는 조이스틱이 움직이지 않는 상태이므로 마우스 커서가 멈추도록 하였고, 중심원의 바깥쪽에 위치할 때는 조이스틱이 그 방향으로 움직인 것이므로 마우스의 커서가 변화점 X가 가리키는 각도의 방향으로 움직이도록 한다.

X 표시가 중심원 안에 있을 때 마우스 커서를 멈추도록 하는 이유는 마우스 포인터가 원하는 지점에 도착했을 때 손의 미세한 떨림에도 포인터가 움직이게 되는 것을 방지하기 위해서다. 화면에 나타나는 중심원의 크기는 지름 100픽셀, 약 3센티미터 정도이며, 통상 1대1에서 3대1까지 매핑이 이루어질 경우, 실제 손의 이동거리는 지름 20~30센티미터가 되고 이 범위 안에서 움직임이 있더라도 조이스틱이 어떤 방향으로도 기울어지지 않은 상태로 인식하게 된다. 만약 중심원을 없애게 되면 마우스 포인터는 계속 떨리게 되며, 손의 제스처로 버튼을 클릭할 경우 포인터가 움직이게 되어 정확한 위치에서의 클릭이 힘들다.

마우스의 새로운 좌표 값을 만들기 위한 방향과 속도를 구하기 위해서는 Fig. 5와 같이 중심점과 변화

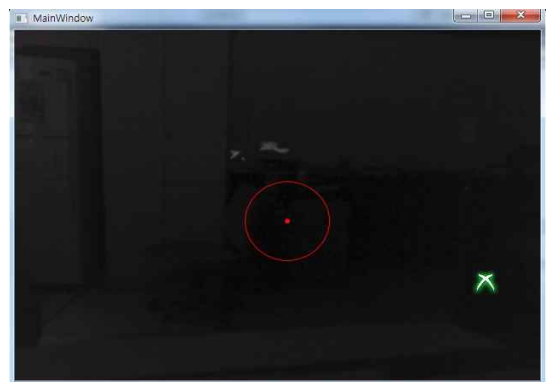


Fig. 4. Output of Kinect display.

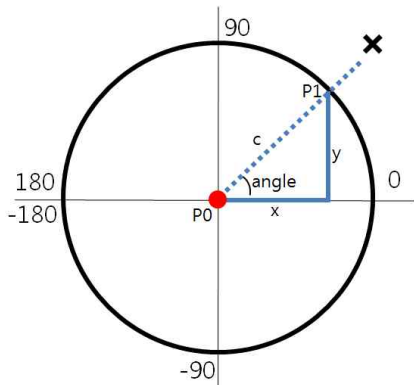


Fig. 5. Mouse control of joystick method.

점 X(손의 위치)사이의 각도와 거리를 구해야 한다. 여기서 사용되는 각도는 호도법, 라디안 각도를 사용하고 라디안 값을 구하는 공식은 식 1과 같다.

$$radian = \text{Atan2}(point.Y - center.Y, point.X - center.X) \quad (1)$$

식 (1)에서 point는 Fig. 5의 변화점 X의 좌표를 나타내고, center는 Fig. 5의 중심점 P0의 좌표를 가리킨다. 위 식에서 구한 radian 값을 Cos(radian) 하면 필요로 하는 좌표의 x좌표가 출력되고 Sin(radian)하면 y좌표가 출력된다. 이 내용을 나타내는 과정은 다음 식 (2)와 같다.

$$\begin{aligned} \text{Cos}(radian) &= x/c \\ \text{Sin}(radian) &= y/c \\ x &= c \times \text{Cos}(radian) \\ y &= c \times \text{Sin}(radian) \end{aligned} \quad (2)$$

위 식 (2)의 c의 값은 마우스 포인터가 이동하는 단위가 되어 마우스의 속도를 제어 할 수 있는 값이 된다. 마우스 포인터가 중심점에서 멀어질수록 속도가 빨라 질 수 있도록 하기 위해 중심점과 손의 위치 X사이의 거리를 구하여 속도상수 c로 나누었다. 속도상수로 나눈 값이 커질수록 속도는 빨라진다. 중심점과 변화점 X 사이의 거리를 이용하여 속도상수를 구하는 식은 다음 식 (3)과 같다.

$$c = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3)$$

식 (2)에서 구해진 x와 y의 값을 mouse_event 함수에 넣어주면 마우스포인터가 손의 방향에 따라 계산된 속도로 움직이게 된다. 손이 중심점에서 멀어질수록 c값이 커지게 되어 빠른 이동이 이루어지며, 중심점에 가까울수록 이동속도는 감소하게 되고, 중심

원 속에 들어오면 이동을 멈춘다. 마우스포인터가 화면의 가장자리까지 이동한 경우는 방향에 따라 모퉁이까지 이동을 계속하며, 마우스포인터가 모퉁이에 도달한 경우, 더 이상 이동하지 않는다.

4. 구현결과 및 분석

4.1 구현결과

본 연구의 구현을 위해 Windows7 운영체제와 Microsoft사의 Kinect for windows와 Kinect SDK 1.8버전, Visual Studio 2010을 이용하였다. 또한 키넥트 SDK에서 제공되는 스켈레톤 트래킹을 이용하여 가상 마우스를 구현하였다. 키넥트 카메라는 사람의 몸을 인식하여 Head, Shoulder, Elbow, Wrist, Hand, Knee, Foot 등 20개의 관절을 인식하지만, 그 중에서 마우스로 사용할 오른손인 Hand_Right를 트래킹 하였고 User32.dll을 이용하여 마우스 포인터가 조이스틱형 방식으로 제어 되도록 하였다. 제어 프로그램에 대한 실행결과 화면은 아래 Fig. 6과 같다.

마우스 버튼의 클릭 부분은 키넥트 SDK의 버전 1.7 이상에서 지원되는 Interaction을 이용하여 오른손을 쥐는 것과 펴는 것을 이용하였다. 오른손을 주먹 쥐면 Grip 상태가 되어 마우스 버튼을 눌린 상태가 되고, 주먹을 펴면 Release 상태가 되어 마우스 버튼을 누르지 않은 상태가 되도록 하였다. 클릭을 하려면 한번 쥐었다 펴면 된다. 이것을 응용하여 주먹을 쥔 채로 손을 움직이면 드래그를 할 수 있도록 하였다. 주먹을 쥐는 동작으로 클릭하는 것에 대한 문제점으로는 더블클릭이 어렵다는 단점이 있었다.



Fig. 6. Video Screen at the Mouse Control.

손을 빠르게 두 번 쥐었다 펴도 잘 인식이 안 될 때가 많아서 이 부분에 대한 개선이 필요하다.

4.2 분석

본 연구의 유효성을 평가하기 위하여 손이 움직이는 만큼만 마우스가 움직이는 마우스-손 동기화 방식과 중심점에서 손이 가리키는 방향과 위치에 따라 움직이게 되는 조이스틱 방식에 대해 30명의 실험자들을 대상으로 성능을 비교 테스트하고 설문을 받아 평가하였다. 평가항목은 정확성, 용이성, 효율성, 편의성으로 나누었다. 정확성은 마우스 커서가 얼마나 정확하고 빠르게 원하는 지점에 도달하는지에 대한 평가이고, 용이성은 사용방법에 대한 설명이 없어도 쉽게 사용하는지에 대한 평가이다. 효율성은 몸의 동작, 움직임에 대한 효율에 대한 분석이고, 편의성은 전체적으로 사용하기 편리한 정도에 대한 측정이다. 평가방식은 실험자 당 각 평가항목 마다 1점부터 5점까지 점수를 체크하여 그 평균치를 구하였고 결과는 Table 1과 같다.

정확성에 대한 결과로 비 조이스틱형은 손을 뺀면 마우스 포인터를 원하는 곳까지 바로 보낼 수 있어 빠르고 정확하지만 화면의 구석으로 마우스 포인터를 보낼 때는 자리를 이동할 정도의 큰 동작이 필요했고 그러한 상황에서는 시간이 조금 더 걸렸다. 조이스틱형도 마찬가지로 원하는 곳까지 빠르게 보낼 수 있었으며, 두 가지 방식 모두 4.3 이상의 만족도를 나타내어 큰 차이가 없지만, 비 조이스틱형이 마우스의 떨림이 없기 때문에 조금 더 정확하다는 설문 결과가 나왔다.

용이성에 대한 결과로는 조이스틱형이 비 조이스틱형에 비해 크게 떨어졌는데 그 이유는 조이스틱형의 원 안과 밖에 대한 기능적인 설명이 없었기 때문이라고 설문 결과 판단하였다. 처음 사용하는 이용자들에게는 약간의 설명과 숙달하고 적응하는데 시간이 필요했다. 하지만 조이스틱형은 중심점과 변화점

에 따라 마우스 포인터가 이동한다는 설명 후, 쉽게 적응하는 모습을 보였다.

효율성에 대한 결과로 PC를 사용하는데 있어 마우스를 장시간 이용하는데 몸을 많이 움직여 쉽게 피로해지는 비 조이스틱형 보다 적게 움직여도 화면 구석까지 커서를 보낼 수 있는 조이스틱형이 훨씬 효율적이라는 결과가 나왔다.

편의성에 대한 결과로 조이스틱형은 원안에 손이 있을 경우 마우스 커서가 그 자리에서 멈추기 때문에 어느 한 지점 위에 커서를 놓기 편했고, 클릭을 하는데 있어서도 커서가 움직이지 않아서 바로 클릭이 가능하다는 장점이 있었다. 반면에 비 조이스틱형은 미세한 손동작에도 반응하여 마우스포인터의 떨림 현상이 발생했으며, 주먹을 쥐는 동작의 클릭 시에 위치에 대한 오인식도 일어났다. 가장자리로의 마우스포인터 이동을 위하여 끝까지 손을 뺐어야 하는 불편함이 있었다.

종합적인 결과로는 조이스틱형이 비 조이스틱형에 비해 첫 사용 시에는 어색하고 사용이 어려울 수 있지만 적응하고 나면 몸이 덜 힘들고 사용하기 편할 것 이라는 결과가 나왔다.

4. 결 론

키넥트 장치는 마우스와 키보드를 대신 할 수 있는 비접촉성 차세대 입력장치로 각광받고 있고 이를 이용하는 다양한 분야의 응용 소프트웨어들이 개발되고 있다. 본 연구에서는 키넥트 SDK의 스키텔론 트래킹을 이용하여 사용자의 손 움직임을 인식시켜 마우스를 제어하였고, 기존의 방식과는 다르게 사용자가 조금의 움직임으로도 좀 더 안정적이고 편안한 조작을 할 수 있기 위해 중심점에서 손바닥이 가리키는 방향과 거리를 계산하여 손이 가리키는 방향으로 마우스 포인터를 이동시키는 조이스틱 방식을 제안하였다. 실험 결과 손바닥의 작은 동작에도 모니터의

Table 1. The Results of Performance Comparison

Section	Case of Non Joystick[5/4/3/2/1]	Case of Joystick [5/4/3/2/1]
Accuracy	4.43 [15/13/ 2/ 0/ 0]	4.30 [12/15/ 3/ 0/ 0]
Easiness of Use	3.96 [10/11/ 7/ 2/ 0]	2.80 [2/ 5/12/ 7/ 4]
Effectiveness	3.03 [3/ 4/12/ 9/ 2]	4.43 [17/ 9/ 4/ 0/ 0]
Convenience	3.53 [4/11/12/ 3/ 0]	3.76 [8/13/ 6/ 3/ 0]

모든 위치까지 빠르고 정확하게 마우스 포인터를 위치시킬 수 있었으며 조이스틱 방식이 아닐 때와 비교하여 효율성이나 편의성이 증가한 것을 알 수 있었다. 다만 사용자가 사용법을 모른 채로 사용하게 되면 조이스틱방식에 적응하여 사용하는데 약간의 시간이 걸린다는 단점이 있음을 확인하였다. 이는 사용법을 미리 알려줌으로써 충분히 해소할 수 있다고 판단된다. 앞으로 왼손의 동작을 함께 인지하여 스위이프, 스크롤, 확대, 축소 등과 같은 이벤트도 처리할 수 있도록 확장해 나갈 계획이다.

REFERENCE

- [1] An Innovative User Interface Implementation of Embedded Device for Kinect(Windows), <http://www.embeddednews.co.kr/news/articleView.html?idxno=5532> (accessed June, 10, 2015).
- [2] Y.J. Youn, S.G. Son, W.A. Jang, and D.H. Lee, "Presentation Control in Pen-mode by Gesture using Kinect," *Proceeding of Conference of the Korean Institute Scientists and Engineers*, Vol. 2013, No. 11, pp. 1117-1119, 2013.
- [3] S.M. Kim, J.K. Song, B.W. Yoon, and J.S. Park, "Height Estimation using Kinect in the Indoor," *Journal of the Korea Institute of Electronic Communication Sciences*, Vol. 9, No. 3, pp. 343-350, 2014.
- [4] M.K. Lee and J.B. Jeon, "Personal Computer Control Using Kinect," *Proceeding of Conference of the Korean Institute Scientists and Engineers*, Vol. 39, No. 1(A), pp. 343-345, 2012.
- [5] S.Y. Youn and K.C. Kwak, "Implementation of Motion-based Interface Using Kinect Sensor," *Proceeding of Korean Institute of Information Technology*, Vol. 2012, No. 5, pp. 393-395, 2012.
- [6] I.B. Jeon and B.H. Nam, "Implementation of Hand Mouse based on Depth Sensor of the Kinect," *Proceeding of Korean Institute of Electrical Engineers*, Vol. 2012, No. 7, pp. 1674-1675, 2012.
- [7] J.I. Kim, "A Mouse Control Method Using Hand Movement Recongnition," *Journal of Korea Multimedia Society*, Vol. 15, No. 11, pp. 1377-1383, 2012.
- [8] S. Rajaganapathy, B. Aravind, B. Keerthana, and M. Sivagami, "Conversation of Sign Language to Speech with Human Gestures," *Proceeding of Computer Science*, No. 50, pp. 10-15, 2015.
- [9] A. Sanna, F. Lamberti, G. Paravati, and F. Manuri, "A Kinect-based Natural Interface for Quadrotor Control," *Entertainment Computing*, Vol 4, pp. 179-186, 2013.
- [10] Y.A. Yusoff, A.H. Basori, and F. Mohamed, "Interactive Hand and Arm Gesture Control for 2D Medical Image and 3D Volumetric Medical Visualization," *Proceeding of Social and Behavioral Sciences*, No. 97, pp. 723-729, 2013.
- [11] J. Webb and J. Ashley, *Beginning Kinect Programming with the Microsoft Kinect SDK*, Apress, New York, 2012.
- [12] Principle of Kinect Sensor 1, <http://minsug.tistory.com/24> (accessed Dec, 14, 2014)
- [13] D.H. Kim, *Creation of Contents Interact with Objet Using Kinect Motion Recognition Sensor*, Master's Thesis of Kyungbook National University, 2012.



정진영

2007년 3월~2014년 3월, 동명대학교 컴퓨터공학과 학사
2014년 3월~현재, 동명대학교 컴퓨터미디어공학과 석사
관심분야: 기계학습, 영상처리, 자바, 빅데이터



김정인

1991년 4월~1993년 3월 게이오대학 계산기과학전공 공학석사
1993년 4월~1996년 3월 게이오대학 계산기과학전공 공학박사

1996년 5월~1998년 2월 포항공과대학교 정보통신연구소 연구원, 기계번역시스템 설계
1998년 3월~현재, 동명대학교 컴퓨터공학과 교수
관심분야: 기계번역, 기계학습, 시멘틱웹, 웹2.0